Lecture Notes in Statistical and Mathematical Methods for Artificial Intelligence 91255

Giorgio Renzi
giorgio.renzi@studio.unibo.it

Academic Year 2019/2020

Last updated: February 2020

Thanks to (in alphabetical order):

Lorenzo Mario Amorosa
Lucia La Forgia
Giacomo Pinardi

for the help provided

Contents

1 Elements of Linear Algebra

1.1 Vector spaces and vectors

Definition 1.1. A *vector space* over a field F ($F = \mathbb{R}$ or $F = \mathbb{C}$) is a set closed under finite *vector addition* and *scalar multiplication*. The elements of V are called *vectors* while the elements of F are called *scalars*. The two operations must satisfy the following properties:

1. Commutativity of addition

$$\forall \mathbf{v}, \mathbf{w} \in V,\ \mathbf{v} + \mathbf{w} = \mathbf{w} + \mathbf{v}$$

2. Associativity of addition

$$\forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in V,\ \mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$$

3. Existence of the identity element of addition, $\mathbf{0} \in V$, such that

$$\forall \mathbf{v} \in V,\ \mathbf{v} + \mathbf{0} = \mathbf{0}$$

4. Existence of the additive inverse

$$\forall \mathbf{v} \in V,\ \exists -\mathbf{v} \in V,\ \mathbf{v} + (-\mathbf{v}) = \mathbf{0}$$

5. Existence of the identity element of scalar multiplication, $1 \in F$, such that

$$\forall \mathbf{v} \in V,\ 1\mathbf{v} = \mathbf{v}$$

6. Existence of the zero element of scalar multiplication, $0 \in F$, such that

$$\forall \mathbf{v} \in V,\ 0\mathbf{v} = \mathbf{0}$$

7. Compatibility of scalar multiplication with field multiplication

$$\forall a, b \in F,\ \forall \mathbf{v} \in V,\ (ab)\mathbf{v} = a(b\mathbf{v})$$

8. Distributivity of scalar multiplication with respect to field addition

$$\forall a, b, \in F,\ \forall \mathbf{v} \in V,\ (a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$$

9. Distributivity of scalar multiplication with respect to vector addition

$$\forall a \in F,\ \forall \mathbf{v}, \mathbf{w} \in V,\ a(\mathbf{v} + \mathbf{w}) = a\mathbf{v} + a\mathbf{w}$$

Example 1.1. Some notable examples of vector spaces are:

- $\mathbb{R}^n$, the set of n-tuples of real numbers
- $\mathbb{C}^n$, the set of n-tuples of complex numbers
- $\mathbb{P}_n$, the set of polynomials having degree less or equal to n
- $C^n([a, b])$, the set of real (or complex)-valued functions continuous on $[a, b]$ up to their n-th derivative, $n \in [0, \infty)$

Definition 1.2. Given V vector space over the field F, the set W is a *subspace* of V if and only if:

- $W \subset V$
- W is a vector space over F

Example 1.2. Given $V = \mathbb{R}^3$ over $\mathbb{R}$, the set

$$W = \{(\alpha, 0, 0),\ \alpha \in \mathbb{R}\}$$

is a subspace of V because

- $\forall \mathbf{w}_1, \mathbf{w}_2 \in W,\ \mathbf{w}_1 + \mathbf{w}_2 \in W$
- $\forall \alpha \in \mathbb{R},\ \forall \mathbf{w} \in W,\ \alpha\mathbf{w} \in W$

Example 1.3. Given $V = \mathbb{R}^3$ over $\mathbb{R}$, the set

$$U = \{(1, 0, 0), (1, 2, 3)\}$$

is not a subset of V because it is not a vector space, since

$$(1, 0, 0) + (1, 2, 3) \notin U$$

1.1.1 Linear independence

Definition 1.3. Given a vector space V over F, the set W of all finite linear combinations of vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_m\},\ \mathbf{v}_i \in V$, is called the *generated*

subspace or *span* of the vector system, and is written as

$$W = \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_m\} = \left\{ \sum_{i=1}^{m} \alpha_i \mathbf{v}_i \;\middle|\; \mathbf{v}_i \in V,\ \alpha_i \in F \right\}$$

The system $\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ is called a *system of generators* of V.

Example 1.4. $\{(-1,0,0),(0,1,0),(0,0,2),(-1,0,4)\}$ is a system of generators for $\mathbb{R}^3$

Definition 1.4. Given a vector space V over F, a system of vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_m\},\ \mathbf{v}_i \in V$ is called *linearly independent* if

$$\alpha_1 \mathbf{v}_1 + \ldots + \alpha_m \mathbf{v}_m = \mathbf{0} \implies \alpha_1 = \alpha_2 = \ldots = \alpha_m = 0$$

with $\alpha_1, \ldots, \alpha_m \in F$. Otherwise, the system is called *linearly dependent*.

From a geometrical point of view, n vectors are linearly dependent if they lie on the same $(n-1)$-dimensional hyperplane.

Example 1.5. Given $V = \mathbb{R}^2,\ \mathbf{v}_1 = (1,2),\ \mathbf{v}_2 = (3,4)$

$$\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 = \mathbf{0} \implies$$

$$\implies \alpha_1(1,2) + \alpha_2(3,4) = (0,0) \implies$$

$$\implies (\alpha_1 + 3\alpha_2, 2\alpha_1 + 4\alpha_2) = (0,0) \implies$$

$$\implies \begin{cases} \alpha_1 + 3\alpha_2 = 0 \\ 2\alpha_1 + 4\alpha_2 = 0 \end{cases} \implies \begin{cases} \alpha_1 = 0 \\ \alpha_2 = 0 \end{cases}$$

Thus, $\mathbf{v}_1$ and $\mathbf{v}_2$ are linearly independent.

Definition 1.5. We call a *basis* of a vector space V any system of linearly independent generators of V.

Example 1.6. $\{(1,0,0),(0,1,0),(0,0,1)\}$ is a basis for $\mathbb{R}^3$.

Proposition 1.1. Let V be a vector space which admits a basis of n vectors. Then every system of linearly independent vectors has at most n elements and any other basis of V has exactly n elements. The number n is called *dimension* of V and is denoted by $\dim(V) = n$.

1.2 Matrices

Definition 1.6. Let m and n be two positive integers. We call *matrix* the rectangular array having m rows and n columns of elements in a field F. It is represented as

$$\mathrm{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

If the field is $F = \mathbb{R}$ (respectively $F = \mathbb{C}$) we write $\mathrm{A} \in \mathbb{R}^{m \times n}$ (respectively $\mathrm{A} \in \mathbb{C}^{m \times n}$). If $m = n$ the matrix is called *square*. The set of entries A_{ij} with $i = j$ is called *main diagonal* of the matrix A.

We can write the above matrix as $A(m \times n)$ or $A = (a_{ij})$, $i = 1, \ldots, m$ and $j = 1, \ldots, n$.

Definition 1.7. Let $\mathrm{A} \in F^{m \times n}$. The maximum number of linearly independent columns (or rows) of A is called *rank*, denoted by rank(A). A is said to have *complete* or *full rank* if $\mathrm{rank(A)} = \min(m, n)$.

Definition 1.8. A *lower triangular matrix* is a square matrix $\mathrm{L}(n \times n)$ where $l_{ij} = 0$ if $i < j$. An *upper triangular matrix* is a square matrix $\mathrm{U}(n \times n)$ where $u_{ij} = 0$ if $i > j$.

$$\mathrm{L} = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix}, \ \mathrm{U} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix}$$

1.2.1 Operations with matrices

Let $\mathrm{A} \in F^{m \times p}$, $\mathrm{B} \in F^{m \times p}$, $\mathrm{C} \in F^{p \times n}$, $\lambda \in F$. We define the following operations:

– *matrix addition*

$$\mathrm{A} + \mathrm{B} = (a_{ij} + b_{ij}), \ \mathrm{A} + \mathrm{B} \in F^{m \times p}$$

The identity element of matrix addition is the *null matrix*, denoted by 0 and made up only by null entries;

– *matrix multiplication by a scalar*

$$\lambda \mathrm{A} = (\lambda a_{ij}),\ \lambda \mathrm{A} \in F^{m \times p}$$

– *matrix multiplication*

$$\mathrm{AC} = \left(\sum_{k=1}^{p} a_{ik} b_{kj} \right),\ \mathrm{AC} \in F^{m \times n}$$

Notice how matrix multiplication is defined only when the number of columns of the first matrix is equal to the number of rows of the second matrix. This operation results in a matrix with size (m, n). Matrix product is associative and distributive with respect to matrix addition, but it is not in general commutative

$$\mathrm{AC} \neq \mathrm{CA}$$

We call *commutative* the square matrices for which $\mathrm{AC} = \mathrm{CA}$ holds.

– *transposition*

$$\mathrm{A}^T = (a_{ji}),\ \mathrm{A}^T \in F^{p \times m}$$

Transposition enjoys the following properties:

$$\left(\mathrm{A}^{\mathrm{T}}\right)^T = A,\ (\mathrm{A} + \mathrm{B})^T = \mathrm{A}^T + \mathrm{B}^T,\ (\mathrm{AC})^T = \mathrm{C}^T \mathrm{A}^T,\ (\lambda \mathrm{A})^T = \lambda \mathrm{A}^T$$

Example 1.7. Let

$$\mathrm{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix},\ \mathrm{B} = \begin{bmatrix} 7 & 8 & 9 \\ 1 & 2 & 3 \end{bmatrix}$$

Then we have

$$\mathrm{A} + \mathrm{B} = \begin{bmatrix} 1+7 & 2+8 & 3+9 \\ 4+1 & 5+2 & 6+3 \end{bmatrix} = \begin{bmatrix} 8 & 10 & 12 \\ 5 & 7 & 9 \end{bmatrix}$$

$$\mathrm{B}^T = \begin{bmatrix} 7 & 1 \\ 8 & 2 \\ 9 & 3 \end{bmatrix}$$

$$\mathrm{AB}^T = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 7 & 1 \\ 8 & 2 \\ 9 & 3 \end{bmatrix} =$$

$$= \begin{bmatrix} 1 \cdot 7 + 2 \cdot 8 + 3 \cdot 9 & 1 \cdot 1 + 2 \cdot 2 + 3 \cdot 3 \\ 4 \cdot 7 + 5 \cdot 8 + 6 \cdot 9 & 4 \cdot 1 + 5 \cdot 2 + 6 \cdot 3 \end{bmatrix} = \begin{bmatrix} 50 & 14 \\ 122 & 32 \end{bmatrix}$$

Definition 1.9. A *diagonal matrix* of order n is a square matrix of the type $A = (d_{ii}\delta_{ij})$, where δ_{ij} denotes the *Kronecker symbol* equal to 1 if $i = j$ and 0 otherwise. It can be written as $A = \text{diag}(d_{11}, d_{22}, \ldots, d_{nn})$ or $A = \text{diag}(\mathbf{d})$, where $\mathbf{d} = (d_{11}, d_{22}, \ldots, d_{nn})$.

Definition 1.10. We call *identity matrix of order n* the square matrix of size (n, n) given by $\mathrm{I}_n = \text{diag}(\mathbf{1})$. Usually, if the order is implied by the context, we write the identity matrix as I.

$$\mathrm{I} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

The identity matrix is, by definition, the only matrix such that $\mathrm{AI}_n = \mathrm{I}_n\mathrm{A} = \mathrm{A}$, for all $\mathrm{A}(n \times n)$.

Definition 1.11. A matrix $\mathrm{A}(n \times n)$ is called *invertible* (or *nonsingular*) if there exists a matrix $\mathrm{B}(n \times n)$ such that $\mathrm{AB} = \mathrm{BA} = \mathrm{I}$. B is called *inverse matrix* of A and is denoted by A^{-1}. A non-invertible square matrix is called *singular*.

The inverse of an invertible matrix is also invertible, $\left(\mathrm{A}^{-1}\right)^{-1} = \mathrm{A}$, and if two square matrices A and B are both invertible, their product is also invertible, $(\mathrm{AB})^{-1} = \mathrm{B}^{-1}\mathrm{A}^{-1}$. Moreover, if a square matrix A is invertible, then $\left(\mathrm{A}^{\mathrm{T}}\right)^{-1} = \left(\mathrm{A}^{-1}\right)^{T} = \mathrm{A}^{-T}$

Proposition 1.2. A square matrix is invertible iff its column vectors are linearly independent.

Example 1.8. Let

$$\mathrm{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Let's check if its column vectors are linearly independent

$$\begin{cases} \alpha_1 + 2\alpha_2 = 0 \\ 3\alpha_1 + 4\alpha_2 = 0 \end{cases} \implies \begin{cases} \alpha_1 = 0 \\ \alpha_2 = 0 \end{cases}$$

Since the column vectors are independent we know that A is invertible. A method to compute the inverse will be given in the next section.

Definition 1.12. A square matrix A is called *symmetric* if $\mathrm{A} = \mathrm{A}^T$, *anti-symmetric* if $\mathrm{A} = -\mathrm{A}^T$, *orthogonal* if $\mathrm{A}^{-1} = \mathrm{A}^T$, i.e. $\mathrm{A}\mathrm{A}^T = \mathrm{A}^T\mathrm{A} = \mathrm{I}$.

Example 1.9. The following matrix is symmetric

$$\mathrm{A} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 4 \\ 3 & 4 & 1 \end{bmatrix}$$

The following matrix is antisymmetric

$$\mathrm{A} = \begin{bmatrix} 0 & 2 & -3 \\ -2 & 0 & -4 \\ 3 & 4 & 0 \end{bmatrix}$$

The following matrix is orthogonal

$$\mathrm{A} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

1.2.2 Determinant of a matrix

Definition 1.13. Let $\mathrm{A} \in F^{n\times n}$. We call the *determinant* of A, denoted by $\det A$, the scalar defined through the following formula

$$\det(\mathrm{A}) = \begin{cases} a_{11} & \text{if } n = 1 \\ \displaystyle\sum_{j=1}^{n} (-1)^{i+j} \det(\mathrm{A}_{ij}) a_{ij} = \sum_{i=1}^{n} (-1)^{i+j} \det(\mathrm{A}_{ij}) a_{ij} & \text{if } n > 1 \end{cases}$$

with $i \in \{1, \ldots, n\}$ (or $j \in \{1, \ldots, n\}$) fixed, where A_{ij} is the submatrix of order $n-1$ obtained from A by eliminating row i and column j. This is known as *Laplace's rule*

If A is diagonal or triangular, then

$$\det(\mathrm{A}) = \prod_{i=1}^{n} a_{ii}$$

The determinant enjoys the following properties

$$\det(\mathrm{A}) = \det(\mathrm{A}^T),\ \det(\mathrm{AB}) = \det(\mathrm{A})\det(\mathrm{B}),\ \det(\mathrm{A}^{-1}) = \frac{1}{\det(\mathrm{A})},$$

$$\det(\alpha\mathrm{A}) = \alpha^n \det(\mathrm{A}),\ \forall \alpha \in F$$

Proposition 1.3. Every orthogonal matrix A is invertible and

$$\det(\mathrm{A}) = \pm 1$$

In the following examples we illustrate methods to compute a matrix determinant that do not require the use of Laplace's rule.

Example 1.10. Let

$$\mathrm{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

The determinant of a 2×2 matrix can be computed as

$$\det(\mathrm{A}) = a_{11}a_{22} - a_{21}a_{12} = 1 \cdot 4 - 3 \cdot 2 = -2$$

Example 1.11. Let

$$\mathrm{A} = \begin{bmatrix} 1 & 2 & 0 \\ 3 & 4 & 5 \\ 3 & 0 & 4 \end{bmatrix}$$

The determinant of a 3×3 matrix can be computed using *Sarrus' rule*

$$\det(\mathrm{A}) = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{21}a_{32}a_{13} - a_{13}a_{22}a_{31} - a_{12}a_{21}a_{33} - a_{23}a_{32}a_{11} =$$

$$= 1 \cdot 4 \cdot 4 + 2 \cdot 5 \cdot 3 + 3 \cdot 0 \cdot 0 - 0 \cdot 4 \cdot 3 - 2 \cdot 3 \cdot 4 - 5 \cdot 0 \cdot 1 = 22$$

Example 1.12. Let

$$\mathrm{A} = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 3 & 4 & 1 & 0 \\ -1 & 3 & 0 & -2 \\ 0 & -2 & -2 & 3 \end{bmatrix}$$

Let's compute the determinant of A with Laplace's rule, fixing $i = 1$ ($|\cdot|$ denotes the determinant)

$$\det(\mathrm{A}) = 1 \cdot \begin{vmatrix} 4 & 1 & 0 \\ 3 & 0 & -2 \\ -2 & -2 & 3 \end{vmatrix} \cdot 1 + (-1) \cdot \begin{vmatrix} 3 & 1 & 0 \\ -1 & 0 & -2 \\ 0 & -2 & 3 \end{vmatrix} \cdot 2 +$$

$$+1 \cdot \begin{vmatrix} 3 & 4 & 0 \\ -1 & 3 & -2 \\ 0 & -2 & 3 \end{vmatrix} \cdot 0 + (-1) \cdot \begin{vmatrix} 3 & 4 & 1 \\ -1 & 3 & 0 \\ 0 & -2 & -2 \end{vmatrix} \cdot 0 =$$

$$= -21 + 18 + 0 + 0 = -3$$

Proposition 1.4. If A is invertible then

$$\mathrm{A}^{-1} = \frac{\mathrm{C}^T}{\det(\mathrm{A})}$$

where C is the *cofactor matrix* having elements $c_{ij} = (-1)^{i+j}\det(\mathrm{A}_{ij})$, called *cofactors*.

Proposition 1.5. For a matrix $\mathrm{A} \in \mathbb{C}^{n \times n}$ the following properties are equivalent:

- A is nonsingular
- $\det(\mathrm{A}) \neq 0$
- $\mathrm{rank}(\mathrm{A}) = n$
- A has linearly independent columns and rows

Example 1.13. Let

$$\mathrm{A} = \begin{bmatrix} 1 & 2 & 0 \\ 3 & 4 & 5 \\ 3 & 0 & 4 \end{bmatrix}$$

We already know (see Example 1.11) that $\det(\mathrm{A}) = 22$. Let's compute the cofactor matrix C

$$\det(\mathrm{A}_{11}) = \begin{vmatrix} 4 & 5 \\ 0 & 4 \end{vmatrix} = 16, \ \det(\mathrm{A}_{12}) = \begin{vmatrix} 3 & 5 \\ 3 & 4 \end{vmatrix} = -3, \ \det(\mathrm{A}_{13}) = \begin{vmatrix} 3 & 4 \\ 3 & 0 \end{vmatrix} = -12$$

$$\det(\mathrm{A}_{21}) = \begin{vmatrix} 2 & 0 \\ 0 & 4 \end{vmatrix} = 8,\ \det(\mathrm{A}_{22}) = \begin{vmatrix} 1 & 0 \\ 3 & 4 \end{vmatrix} = 4,\ \det(\mathrm{A}_{23}) = \begin{vmatrix} 1 & 2 \\ 3 & 0 \end{vmatrix} = -6$$

$$\det(\mathrm{A}_{31}) = \begin{vmatrix} 2 & 0 \\ 4 & 5 \end{vmatrix} = 10,\ \det(\mathrm{A}_{32}) = \begin{vmatrix} 1 & 0 \\ 3 & 5 \end{vmatrix} = 5,\ \det(\mathrm{A}_{33}) = \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} = -2$$

So we have

$$\mathrm{C} = \begin{bmatrix} 16 & 3 & -12 \\ -8 & 4 & 6 \\ 10 & -5 & -2 \end{bmatrix}$$

Thus,

$$\mathrm{A}^{-1} = \frac{\mathrm{C}^T}{\det(\mathrm{A})} = \begin{bmatrix} \frac{8}{11} & -\frac{4}{11} & \frac{5}{11} \\ \frac{3}{22} & \frac{2}{11} & -\frac{5}{22} \\ -\frac{6}{11} & \frac{3}{11} & -\frac{1}{11} \end{bmatrix}$$

1.2.3 Eigenvalues and eigenvectors

Definition 1.14. Let $\mathrm{A} \in \mathbb{C}^{n\times n}$. The number $\lambda \in \mathbb{C}$ is called an *eigenvalue* of A if

$$\exists \mathbf{x} \in \mathbb{C}^n,\ \mathbf{x} \neq \mathbf{0} \text{ such that } \mathrm{A}\mathbf{x} = \lambda\mathbf{x}$$

The vector $\mathbf{x}$ is called the *eigenvector* associated with the eigenvalue λ and the set of eigenvalues of A is called *spectrum* of A and is denoted by $\sigma(\mathrm{A})$. The eigenvalues of A are the solutions of the *characteristic equation*

$$p_{\mathrm{A}}(\lambda) = \det(\mathrm{A} - \lambda\mathrm{I}) = 0$$

Where $p_{\mathrm{A}}(\lambda)$ is called *characteristic polynomial*. For the fundamental theorem of algebra the characteristic polynomial, which has degree n, has exactly n solutions. Hence, a matrix with real or complex entries has n eigenvalues, counted with their multiplicity.

Definition 1.15. The maximum module of the eigenvalues of a matrix $\mathrm{A} \in \mathbb{C}^{n\times n}$ is called the *spectral radius* of A and is denoted by

$$\rho(\mathrm{A}) = \max_{\lambda\in\sigma(\mathrm{A})} |\lambda|$$

Proposition 1.6. It can be proven that for a matrix $\mathrm{A} \in \mathbb{C}^{n\times n}$ with eigenvalues $\lambda_1, \ldots, \lambda_n$

$$\det(\mathrm{A}) = \prod_{i=1}^{n} \lambda_i$$

Proposition 1.7. A matrix is singular if it has at least one null eigenvalue.

Proposition 1.8. Let $\mathrm{A} \in \mathbb{C}^{n\times n}$ with eigenvalues $\lambda_1, \ldots, \lambda_n$. If A is diagonal or triangular, then $\lambda_i = a_{ii}$, $i = 1, \ldots, n$.

1.2.4 Similarity

Definition 1.16. Two matrices $\mathrm{A} \in \mathbb{C}^{n\times n}$ and $\mathrm{B} \in \mathbb{C}^{n\times n}$ are called *similar* if they have the same eigenvalues.

Definition 1.17. Let P be a nonsingular square matrix having the same order as the matrix A. We say that the matrices A and $\mathrm{B} = \mathrm{PAP}^{-1}$

1.3 Scalar Product and Norms in Vector Spaces

Definition 1.18. A *scalar product* on a vector space V over the field F is a mapping $\langle \cdot, \cdot \rangle : \ V \times V \to F$ which enjoys the following properties:

- $\langle \lambda\mathbf{x} + \gamma\mathbf{y}, \mathbf{z}\rangle = \lambda\langle \mathbf{x}, \mathbf{z}\rangle + \gamma\langle \mathbf{y}, \mathbf{z}\rangle, \ \forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in V, \ \forall \lambda, \gamma \in F$
- $\langle \mathbf{x}, \mathbf{y}\rangle = \overline{\langle \mathbf{y}, \mathbf{x}\rangle}, \ \forall \mathbf{x}, \mathbf{y} \in V$
- $\langle \mathbf{x}, \mathbf{x}\rangle \geq 0, \ \forall \mathbf{v} \in V$ and $\langle \mathbf{x}, \mathbf{x}\rangle = 0 \iff \mathbf{x} = \mathbf{0}$

A notable example of scalar product is the *Euclidean scalar product*

$$\langle \mathbf{x}, \mathbf{y}\rangle = \sum_{i=1}^{n} x_i y_i, \ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$$

Definition 1.19. Let V be a vector space over the field F. We define the map $||\cdot|| : V \to F$ as a *norm* on V if the following properties are satisfied:

- $||\mathbf{v}|| \geq 0, \ \forall \mathbf{v} \in V$ and $||\mathbf{v}|| = 0 \iff \mathbf{v} = \mathbf{0}$
- $||\alpha\mathbf{v}|| = |\alpha|||\mathbf{v}||, \ \forall \alpha \in F, \ \forall \mathbf{v} \in V$
- $||\mathbf{v} + \mathbf{w}|| \leq ||\mathbf{v}|| + ||\mathbf{w}||, \ \forall \mathbf{v}, \mathbf{w} \in V$

where $|\alpha|$ denotes the absolute value of α if $F = \mathbb{R}$, the module of α if $F = \mathbb{C}$. $(V, ||\cdot||)$ is called a *normed space*.

Example 1.14. Let $\mathbf{v} = (1, 2, 3)$. The Euclidean norm of $\mathbf{v}$ is

$$||\mathbf{v}||_2 = \sqrt{v_1^2 + v_2^2 + v_3^2} = \sqrt{14}$$

Example 1.15. Let $\mathbf{v} = (1, 2, 3)$. The *Manhattan norm* of $\mathbf{v}$ is

$$||\mathbf{v}||_1 = \sum_{i=1}^{n} |v_i| = 6$$

Example 1.16. Let $\mathbf{v} \in \mathbb{R}^n$. The *p-norm* of $\mathbf{v}$ is

$$||\mathbf{v}||_p = \left(\sum_{i=1}^{n} |v_i|^p \right)^{1/p}, \ 1 \leq p < \infty$$

Notice how the p-norm with $p = 1$ is the Manhattan norm and with $p = 2$ is the Euclidean norm.

Example 1.17. Let $\mathbf{v} = (-8, 2, 5)$. The *infinity norm* of $\mathbf{v}$ is

$$||\mathbf{v}||_\infty = \max_{i=1,\ldots,n} |v_i| = 8$$

1.3.1 Matrix norms

Definition 1.20. A *matrix norm* is a map $||\cdot|| : \mathbb{R}^{m\times n} \to \mathbb{R}$ satisfying the following properties:

- $||\mathrm{A}|| \geq 0, \ \forall \mathrm{A} \in \mathbb{R}^{m\times n}$ and $||\mathrm{A}|| = 0 \iff \mathrm{A} = 0$
- $||\alpha \mathrm{A}|| = |\alpha| ||\mathrm{A}||, \ \forall \alpha \in \mathbb{R}, \ \forall \mathrm{A} \in \mathbb{R}^{m\times n}$
- $||\mathrm{A} + \mathrm{B}|| \leq ||\mathrm{A}|| + ||\mathrm{B}||, \ \forall \mathrm{A}, \mathrm{B} \in \mathbb{R}^{m\times n}$

Definition 1.21. Let $\mathrm{A} \in \mathbb{R}^{n\times n}$. We say the norm $||\cdot||$ is compatible with a vector norm $||\cdot||$ if

$$||\mathrm{A}\mathbf{x}|| \leq ||\mathrm{A}|| ||\mathbf{x}||$$

Example 1.18. Let $\mathrm{A} \in \mathbb{R}^{m \times n}$. The *spectral norm* of A is

$$||\mathrm{A}||_2 = \sqrt{\rho(\mathrm{A}^T A)}$$

The spectral norm of the identity matrix is

$$||\mathrm{I}||_2 = 1$$

Example 1.19. Let $\mathrm{A} \in \mathbb{R}^{m \times n}$. The *p-norm* of A with $p = 1$ is

$$||\mathrm{A}||_1 = \max_{j=1,\ldots,n} \sum_{i=1}^{n} |a_{ij}|$$

The 1-norm of the identity matrix is

$$||\mathrm{I}||_1 = 1$$

Example 1.20. Let $\mathrm{A} \in \mathbb{R}^{m \times n}$. The *infinity norm* of A is

$$||\mathrm{A}||_\infty = \max_{i=1,\ldots,n} \sum_{j=1}^{n} |a_{ij}|$$

Notice how if A is symmetric $||\mathrm{A}||_1 = ||\mathrm{A}||_\infty$. The infinity norm of the identity matrix is

$$||\mathrm{I}||_\infty = 1$$

Example 1.21. Let $\mathrm{A} \in \mathbb{R}^{m \times n}$. The *Frobenius norm* of A is

$$||\mathrm{A}||_F = \sqrt{\sum_{i,j=1}^{n} |a_{ij}|^2}$$

The Frobenius norm of the identity matrix is

$$||\mathrm{I}_n||_F = \sqrt{n}$$

2 Numerical Computation and Finite Numbers

Without going into details, we define *numerical method* a mathematical tool designed to solve numerical problems. The implementation of a numerical method is called a *numerical algorithm*, or simply *algorithm*.

When working with numerical methods we can incour in a series of approximations and errors which we can classify in the following way:

- *measure errors*, caused by the measuring instrument;
- *arithmetic (or algorithmic) errors*, caused by the propagation of rounding errors of each operation in an algorithm;
- *truncation errors*, caused by the truncation of an infinite procedure to a finite one (e.g. the approximation of an infinite series with a finite sum);
- *propagation (or inherent) errors*, caused by the finite representation of the data of a problem

Definition 2.1. Let $\tilde{x}$ be an approximation of a datum x. We define the *absolute error* as

$$E_x = |x - \tilde{x}|$$

and the *relative error* as

$$R_x = \left| \frac{x - \tilde{x}}{x} \right|, \ x \neq 0$$

The primary goal of a numerical method is to make these errors arbitrarily small. Another goal is *accuracy*.

Definition 2.2. *Accuracy* is the number of correct significant digits in approximating some quantity. Accuracy means that the errors are small relative to a fixed tolerance. This quantity is not limited by the machine *precision*, which is the number of digits a number is expressed with.

Definition 2.3. The number $\tilde{x}$ is said to approximate x to d significant digits if d is the largest non-negative integer such that

$$\left| \frac{x - \tilde{x}}{x} \right| < \frac{10^{1-d}}{2}$$

Example 2.1. Let $x = 3.141592$, $\tilde{x} = 3.14$. Then we have

$$\left|\frac{x - \tilde{x}}{x}\right| = 0.000507 < \frac{10^{\mathbf{1-3}}}{2} = 0.5 \cdot 10^{-2}$$

Then we say that $\tilde{x}$ approximates x to $d = 3$ significant digits.

Suppose we want to compute the value of a function $f : \mathbb{R} \to \mathbb{R}$, where:

- x is the true input value
- $f(x)$ is the desired (true) output
- $\tilde{x}$ is the approximate input
- $\tilde{f}$ is the approximate computed function

Then we define the *total error* as

$$\tilde{f}(\tilde{x}) - f(x) = \tilde{f}(\tilde{x}) - f(\tilde{x}) + f(\tilde{x}) - f(x)$$

We call $\tilde{f}(\tilde{x}) - f(\tilde{x})$ the *computational error* and $f(\tilde{x}) - f(x)$ the *propagated data error*. It can be seen that the choice of algorithm has no effect on the propagated data error.

In light of this new perspective, we can define the truncation and rounding errors.

Definition 2.4. The *truncation error* is the difference between the true result (for the actual input) and the result produced by the given algorithm using exact arithmetic. The *rounding error* is the difference between the result produced by the given algorithm using exact arithmetic and the result given by the same algorithm using limited precision arithmetic (due to inexact representation of real numbers and operations upon them).

The *computational error* is a sum of these two errors (one generally prevails over the other).

2.1 Machine representation of number

Let $\beta \in \mathbb{N}$ be fixed with $\beta \geq 2$, and $x \in \mathbb{R}$ with a finite number of digits x_k, with $0 \leq x_k < \beta$ for $k = -m, \ldots, n$. The *positional representation* of x with respect to the base β is

$$x_\beta = \text{sign}(x)(x_n\beta^n + x_{n-1}\beta^{n-1} + \ldots + x_1\beta + x_0 + x_{-1}\beta^{-1} + \ldots + x_{-m}\beta^{-m}),\ x_n \neq 0$$

where

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \end{cases}$$

Another way to express real numbers is the *floating-point representation*, given by:

$$x = \text{sign}(x) \cdot (a_0.a_1 \dots a_{t-1}) \cdot \beta^e = \text{sign}(x) \cdot m \cdot \beta^{e-t+1}$$

The integer number $m = a_0 a_1 \dots a_{t-1}$, $(0 \leq a_i \leq \beta - 1)$ is called *mantissa* and is such that $0 \leq m \leq \beta^t - 1$. The integer number e is called *exponent* and t is the number of allowed significant digits a_i.

A system of floating-point numbers depends on the following parameters:

- β, base
- t, precision (number of significant digits)
- $[L, U]$, exponent range

It is denoted by

$$\mathcal{F}(\beta, t, L, U) = \{0\} \cup \left\{ x \in \mathbb{R} : x = \text{sign}(x)\beta^e \sum_{i=0}^{t-1} x_i \beta^{-i} \right\}$$

To guarantee uniqueness in the representation of numbers and store one less bit, it is assumed that the leading bit is different than 0 (i.e. $a_0 \neq 0$). This is called *normalized representation*. Therefore, the mantissa is in the range $[\beta^{t-1}, \beta^t - 1]$. The part of the mantissa that is actually stored by a machine (i.e. $a_1 a_2 \dots a_{t-1}$) is called *fraction*.

Example 2.2. The layout of a 32-bit floating point number

$$0\ 01111100\ 01000000000000000000000 = 0.15625$$

We have:

- sign $= 0$ (i.e. it's a positive number)
- $e = -127 + 124 = -3$

– $m = 10100000000000000000000$ (including the hidden bit)

We can convert from floating point representation to the number in base 10 with in the following way:

$$(1.01)_2 \cdot 2^{-3} = (1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2}) \cdot 2^{-3} = 0.15625$$

The total number of normalized floating-point numbers is

$$2(\beta - 1)\beta^{t-1}(U - L + 1) + 1$$

The smallest positive normalized number is

$$\text{UFL} = \beta^L$$

while the largest normalized number is

$$\text{OFL} = (\beta^t - 1)\beta^{U-t+1} = (1 - \beta^{-t})\beta^{U+1}$$

The approximation of a real number x to a floating-point number $fl(x)$ can be achieved by applying one of the following *rounding rules*:

– **round-by-chop**: truncate the base-β expansion of x after t digits (also called round toward zero)

– **round-to-nearest**: $fl(x)$ is set to the nearest floating-point number to x; when there is a tie, the floating-point number whose last digit is even is used

Definition 2.5. Accuracy of floating-point systems is characterized by the *roundoff unit* (or machine precision, or machine epsilon) ϵ_{mach}. When using round-by-chop we have $\epsilon_{mach} = \beta^{1-t}$, while when using round-to-nearest we have $\epsilon_{mach} = \frac{1}{2}\beta^{1-t}$.

Definition 2.6. Another definition of roundoff unit is

$$fl(1 + \epsilon_{mach}) > 1$$

Proposition 2.1. The maximum relative error in representing a real number x within the range of a floating-point system is given by

$$\left| \frac{fl(x) - x}{x} \right| \leq \epsilon_{mach}$$

Proof. Let's define $x \in \mathbb{R}$ as $\text{sign}(x)(d_0.d_1d_2 \dots d_{t-1}d_td_{t+1} \dots)\beta^e$ and $fl(x) = \text{sign}(x)(d_0.d_1d_2 \dots \tilde{d}_{t-1})\beta^e$. We have:

$$|x - fl(x)| = |(d_{t-1} - \tilde{d}_{t-1}).d_t \dots \cdot \beta^{e-t+1}| \leq \frac{1}{2}\beta^{e-t+1}$$

since $|d_{t-1} - \tilde{d}_{t-1}| \leq \frac{1}{2}$, and

$$|x| \geq \beta^e$$

Thus,

$$\left|\frac{x - fl(x)}{x}\right| \leq \frac{1}{2}\beta^{1-t}$$

□

3 Linear Systems

A linear system can be written as

$$\mathrm{A}\mathbf{x} = \mathbf{b}$$

where A is a square matrix of dimension $n \times n$ and $\mathbf{x}$ and $\mathbf{b}$ are column vectors of dimension n. $\mathbf{x}$ represents the unknown solution while $\mathbf{b}$ is a given vector. This form can be expanded as

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned}$$

We are interested in:

- Existence and uniqueness of a solution
- Numerical methods to find the solution
- Conditioning of the problem

Proposition 3.1. The solution of a linear system

$$\mathrm{A}\mathbf{x} = \mathbf{b}$$

with A of dimension $n \times n$, $\mathbf{x}$ and $\mathbf{b}$ of dimension n exists and is unique iff one of the following conditions holds:

– A is non-singular

– $\text{rank}(\text{A}) = n$

The system $\text{A}\mathbf{x} = \mathbf{0}$ admits only the solution $\mathbf{x} = \mathbf{0}$

The solution can be algebrically computed in the following way

$$\text{A}\mathbf{x} = \mathbf{b} \implies \text{A}^{-1}\text{A}\mathbf{x} = \text{A}^{-1}\mathbf{b} \implies \mathbf{x} = \text{A}^{-1}\mathbf{b}$$

The problem with this kind of method is that computing the inverse of A can be expensive for large values of n.

Another method to compute the solution of a system is *Cramer's rule*:

$$x_i = \frac{\det A_i}{\det A}, \ i = 1, \dots, n$$

where A_i is obtained from A by replacing the i-th column by $\mathbf{b}$

As for the previous one, the computation of determinants can be quite expensive for large values of n.

We will consider two different approaches to find the solution of a linear system:

– *direct methods*, if they yield the solution in a finite number of steps; they are more precise but more expensive in terms of computational cost

– *iterative methods*, if they require (in principle) an infinite number of steps; they are less precise and less expensive

3.1 Direct methods

3.1.1 Gaussian Elimination Method and LU factorization

Definition 3.1. Let $\text{A} \in \mathbb{R}^{n \times n}$. Assume that there exist two suitable matrices L and U, lower triangular and upper triangular, such that

$$\text{A} = \text{LU}$$

We call this an LU-*factorization* of A.

If A is non-singular, so are both L and U and the solution of the linear system $\mathrm{A}\mathbf{x} = \mathbf{b}$ can be computed by solving two triangular systems

$$\mathrm{L}\mathbf{y} = \mathbf{b} \text{ (forward substitutions algorithm)}$$
$$\mathrm{U}\mathbf{x} = \mathbf{y} \text{ (backward substitutions algorithm)}$$

Let $\mathrm{A}^{(1)} = \mathrm{A}$ and $k = 1, \ldots, n-1$. The matrix U can be computed as follows:

1. We define $\mathrm{M}^{(k)}$ the matrix of multipliers with elements
   - $m_{ii} = 1,\ i = 1, \ldots, n$
   - $m_{ik} = -\dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}},\ i = k+1, \ldots, n$
   - $m_{ij} = 0,\ otherwise$
2. We compute $\mathrm{A}^{(k+1)} = \mathrm{M}^{(k)}\mathrm{A}^{(k)}$
3. We repeat steps 1-2 until we obtain $\mathrm{A}^{(n)} = \mathrm{U}$

We therefore have

$$\mathrm{M}^{(n-1)}\mathrm{M}^{(n-2)} \ldots \mathrm{M}^{(1)}\mathrm{A} = \mathrm{U}$$
$$\mathrm{A} = \left(\mathrm{M}^{(\mathrm{n}-1)}\mathrm{M}^{(\mathrm{n}-2)} \ldots \mathrm{M}^{(1)}\right)^{-1}\mathrm{U}$$
$$\mathrm{L} = \left(\mathrm{M}^{(\mathrm{n}-1)}\mathrm{M}^{(\mathrm{n}-2)} \ldots \mathrm{M}^{(1)}\right)^{-1}$$

This procedure is called Gaussian Elimination Method and has a computational complexity of $\mathrm{O}(n^3)$. We can then solve the two triangular systems; each substitution has a computational complexity of $\mathrm{O}(n^2)$.

One problem of GEM is that it is *unstable*, i.e. the algorithmic error is not limited. This happens because the elements $a_{kk}^{(k)}$ used to compute the multipliers can be very small or even zero, leading to errors. To avoid this, the *pivoting technique* is employed in order to make the elements $a_{kk}^{(k)}$ as big as possible. This is done by swapping two rows so that $a_{kk}^{(k)}$ is the column element with the maximum absolute value. The swapping can be seen as a left multiplication by a *permutation matrix* P, i.e. an identity matrix with the needed rows swapped. The system becomes:

$$\mathrm{PA}\mathbf{x} = \mathrm{P}\mathbf{b}$$

Definition 3.2. A matrix $\mathrm{A} \in \mathbb{R}^{n \times n}$ is positive semi-definite if

$$\forall \mathbf{x} \in \mathbb{R}^n, \ \mathbf{x} \neq \mathbf{0} \quad \mathbf{x}^T \mathrm{A} \mathbf{x} \geq 0$$

Proposition 3.2. Let $\mathrm{A} \in \mathbb{R}^{n \times n}$ be symmetric positive semi-definite. Then it is always possible to LU-factorize it without using pivoting. In this case we have

$$\mathrm{A} = \mathrm{L}\mathrm{L}^T$$

and the factorization has a computational complexity of $\mathrm{O}(n^3/6)$ (using Cholesky algorithm).

3.2 Iterative methods

Iterative methods are numerical methods that approximate a solution using a procedure involving several (or an infinite amount of) steps. At each step they perform the same operations and compute the residual of the system

$$\mathbf{r}_k = \mathrm{A}\mathbf{x}_k - \mathbf{b}$$

Each iteration requires n^2 multiplications, meaning that these methods have a computational cost of $\mathrm{O}(n^2)$.

The basic idea of iterative methods is to construct a sequence of vectors $\mathbf{x}_k$ that enjoy the property of *convergence*

$$\mathbf{x} = \lim_{k \to \mathrm{inf}} \mathbf{x}_k$$

where $\mathbf{x}$ is the exact solution and the starting guess $\mathbf{x}_0$ is given. The iterative process is stopped at the minimum value of n such that $||\mathbf{x}_n - \mathbf{x}|| < \tau$ where τ is a fixed tolerance and $|| \cdot ||$ is a convenient vector norm.

Iterative methods are sensitive to both algorithmic and truncation errors.

Iterative methods differ from one another in the update function used at each step. There are two main cases:

– *stationary iterative methods*, they take the form

$$\mathbf{x}_{k+1} = \mathrm{B}\mathbf{x}_k + \mathbf{f}$$

where B is called *iteration matrix* and $\mathbf{f}$ is a vector obtained from $\mathbf{b}$

– *conjugate gradient methods*, they take the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

where $\alpha_k \in \mathbb{R}$, $\mathbf{p}_k$ is a vector (usually $\mathbf{p}_k = -\nabla\Phi(\mathbf{x}_k)$, which yields the *gradient method*, also called *steepest descent method*) and A is symmetric positive definite (so that the method converges for every choice of $\mathbf{x}_0$)

3.3 Inherent errors in linear systems

Definition 3.3. The *condition number* of a matrix $\mathrm{A} \in \mathbb{C}^{n\times n}$ is defined as

$$K(\mathrm{A}) = ||\mathrm{A}||\ ||\mathrm{A}^{-1}||$$

where $||\cdot||$ is a matrix norm. In general $K(\mathrm{A})$ depends on the choice of the norm, indicated by a subscript.

Example 3.1. The condition number for p-norm with $p = 2$ is

$$\begin{aligned} ||\mathrm{A}||_2 &= \sqrt{\rho(\mathrm{A}^T\mathrm{A})} = \lambda_{max} \\ ||\mathrm{A}^{-1}||_2 &= \frac{1}{\sqrt{\rho(\mathrm{A}^T\mathrm{A})}} = \frac{1}{\lambda_{min}} \\ K_2(\mathrm{A}) &= ||\mathrm{A}||\ ||\mathrm{A}^{-1}|| = \frac{\lambda_{max}}{\lambda_{min}} \end{aligned}$$

where λ_{max} and λ_{min} are the maximum and minimum eigenvalues of A

Notice that $K(\mathrm{A}) \geq 1$ since

$$1 = ||\mathrm{A}\mathrm{A}^{-1}|| \leq ||\mathrm{A}||\ ||\mathrm{A}^{-1}|| = K(\mathrm{A})$$

Consider now a linear system $\mathrm{A}\mathbf{x} = \mathbf{b}$. What happens to the solution $\mathbf{x}$ when A and/or $\mathbf{b}$ slightly change (e.g. due to machine precision)?

Suppose that A doesn't change while $\mathbf{b}$ changes in $\mathbf{b}+\Delta\mathbf{b}$. The linear system becomes

$$\mathrm{A}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b},\ \Delta\mathbf{x} \text{ inherent error}$$

We want to compare $\left|\left|\dfrac{\Delta\mathbf{x}}{\mathbf{x}}\right|\right|$ and $\left|\left|\dfrac{\Delta\mathbf{b}}{\mathbf{b}}\right|\right|$ to see how much the solution has changed with respect to $\mathbf{b}$. Let's subtract $\mathrm{A}\mathbf{x} = \mathbf{b}$ to $\mathrm{A}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b}$

$$\mathrm{A}\Delta\mathbf{x} = \Delta\mathbf{b} \implies \Delta\mathbf{x} = \mathrm{A}^{-1}\Delta\mathbf{b}$$

Then we have

$$||\Delta \mathbf{x}|| = ||\mathrm{A}^{-1}\Delta \mathbf{b}|| \leq ||\mathrm{A}^{-1}||\ ||\Delta \mathbf{b}||$$

and from the linear system equation

$$||\mathrm{A}\mathbf{x}|| = ||\mathbf{b}|| \implies ||\mathbf{b}|| = ||\mathrm{A}\mathbf{x}|| \leq ||\mathrm{A}||\ ||\mathbf{x}|| \implies ||\mathbf{x}|| \geq \left|\left|\frac{\mathbf{b}}{\mathrm{A}}\right|\right|$$

Thus we have

$$\left|\left|\frac{\Delta \mathbf{x}}{\mathbf{x}}\right|\right| \leq ||\mathrm{A}^{-1}|| \cdot ||\mathrm{A}|| \cdot \frac{||\Delta \mathbf{b}||}{||\mathbf{b}||} = K(\mathrm{A})\frac{||\Delta \mathbf{b}||}{||\mathbf{b}||}$$

The condition number measures how sensitive a function is to changes/errors in the input and how much the output changes as a result of the operations performed. A *well-conditioned* system is a system where $K(\mathrm{A})$ is small, while an *ill-conditioned* system is a system where $K(\mathrm{A})$ is large.

If $K(\mathrm{A})$ is very large, the matrix A is near (with respect to a norm) a singular matrix and its columns are quasi-linearly dependent. Regularization techniques can be used in order to reduce $K(\mathrm{A})$.

4 Orthogonality and Orthogonal Projections

4.1 Orthogonality

Definition 4.1. Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$. $\mathbf{u}$ and $\mathbf{v}$ are *orthogonal* if

$$\langle \mathbf{u}, \mathbf{v} \rangle = 0$$

Definition 4.2. $\mathbf{u} \in \mathbb{R}^n$ is a *unit vector* if $||\mathbf{u}|| = 1$.

Proposition 4.1. (Normalization) $\forall \mathbf{u} \in \mathbb{R}^n, \quad \hat{\mathbf{u}} = \dfrac{\mathbf{u}}{||\mathbf{u}||}$ is a unit vector.

Example 4.1.

$$\mathbf{u} = [1, 2, 3]$$

$$||\mathbf{u}||_2 = \sqrt{14}$$

$$\hat{\mathbf{u}} = \frac{\mathbf{u}}{||\mathbf{u}||} = \left[\frac{1}{\sqrt{14}}, \frac{2}{\sqrt{14}}, \frac{3}{\sqrt{14}}\right]$$

Definition 4.3. The set $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p\},\ \mathbf{u}_i \in \mathbb{R}^n,\ i = 1, \ldots, p$ is an *orthogonal set* if $\langle \mathbf{u}_i, \mathbf{u}_j \rangle = \mathbf{u}_i{}^T \mathbf{u}_j = 0,\ \forall i \neq j$

Example 4.2.

$$\mathbf{u}_1 = [3, 1, 1]^T,\ \mathbf{u}_2 = [-1, 2, 1]^T,\ \mathbf{u}_3 = \left[-\frac{1}{2}, -2, \frac{7}{2}\right]^T$$

is an orthogonal set because

$$\langle \mathbf{u}_1, \mathbf{u}_2 \rangle = 0,\ \langle \mathbf{u}_1, \mathbf{u}_3 \rangle = 0,\ \langle \mathbf{u}_2, \mathbf{u}_3 \rangle = 0,$$

Proposition 4.2. If $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p \in \mathbb{R}^n$ are orthogonal then they are linearly independent. The set $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p\},\ \mathbf{u}_i \in \mathbb{R}^n,\ i = 1, \ldots, p$ is an *orthogonal basis* for $U = \text{span}\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p\}$.

Definition 4.4. The set $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p\},\ \mathbf{u}_i \in \mathbb{R}^n,\ i = 1, \ldots, p$ is an *orthonormal set* if it is an orthogonal set of unit vectors. A basis of orthonormal vectors is called *orthonormal basis*.

Definition 4.5. Let $\mathrm{U} \in \mathbb{R}^{m \times n}$. U is an *orthogonal matrix* iff $\mathrm{U}^T\mathrm{U} = \mathrm{I}$. If $m = n$ then $\mathrm{U}^T = \mathrm{U}^{-1}$.

Proposition 4.3. If $\mathrm{U}^{m \times n}$ is orthogonal then

- $||\mathrm{U}\mathbf{x}||_2 = ||\mathbf{x}||_2,\ \forall \mathbf{x} \in \mathbb{R}^n$
- $\langle \mathrm{U}\mathbf{x}, \mathrm{U}\mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle,\ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$
- $\langle \mathrm{U}\mathbf{x}, \mathrm{U}\mathbf{y} \rangle = 0 \iff \langle \mathbf{x}, \mathbf{y} \rangle = 0,\ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

Transformations by orthogonal matrices preserve both length and angles.

4.2 Projections

Projections are an important class of linear transformations. In machine learning one often deals with high-dimensional data which is hard to visualize and analyze. Oftentimes, only a few dimensions contain useful information, meaning that other dimensions are not essential to understanding the data. When applying data compression techniques we want to minimize the loss of information by finding the most informative dimensions of the data first.

Definition 4.6. Let W be a subspace of $\mathbb{R}^n$ and $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p\}$, $\mathbf{u}_i \in \mathbb{R}^n$, $i = 1, \ldots, p$ an orthogonal basis of W. A vetor $\mathbf{y} \in \mathbb{R}^n$ can be written as

$$\mathbf{y} = \hat{\mathbf{y}} + \mathbf{z}$$

where

$$\hat{\mathbf{y}} = \frac{\mathbf{y}^T\mathbf{u}_1}{||\mathbf{u}_1||^2}\mathbf{u}_1 + \frac{\mathbf{y}^T\mathbf{u}_2}{||\mathbf{u}_2||^2}\mathbf{u}_2 + \ldots + \frac{\mathbf{y}^T\mathbf{u}_p}{||\mathbf{u}_p||^2}\mathbf{u}_p$$

is called the projection of $\mathbf{y}$ on the subspace W. $\mathbf{z}$ is orthogonal to $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p\}$, so $\langle \mathbf{z}, \mathbf{u}_i \rangle = 0$, $i = 1, \ldots, p$. We can write $\mathbf{z} \in W^\perp$ (read as: W orthogonal).

Example 4.3. (In 2 dimensions)
Let $\mathbf{y} \in \mathbb{R}^2$ and $L = \text{span}\{\mathbf{u}\}$ one-dimensional subspace of $\mathbb{R}^2$ generated by $\mathbf{u} \in \mathbb{R}^2$. We can write $\mathbf{y}$ as

$$\mathbf{y} = \hat{\mathbf{y}} + \mathbf{z}$$

and notice that $\mathbf{u} \parallel \hat{\mathbf{y}}$.

The projection $\hat{\mathbf{y}}$ is the closest vector lying in the subspace L to $\mathbf{y}$, where distance is measured by $||\mathbf{y} - \hat{\mathbf{y}}||$. It follows that $\mathbf{z} = \mathbf{y} - \hat{\mathbf{y}}$ is orthogonal to L and, in particular, to $\mathbf{u}$, i.e. $\langle \mathbf{z}, \mathbf{u} \rangle = 0$.

Since $\hat{\mathbf{y}}$ is an element of L it is a multiple of $\mathbf{u}$

$$\hat{\mathbf{y}} = \alpha\mathbf{u}$$

We can find α in the following way

$$\langle \mathbf{z}, \mathbf{u} \rangle = 0 \implies \langle \mathbf{y} - \hat{\mathbf{y}}, \mathbf{u} \rangle = 0 \implies \langle \mathbf{y} - \alpha\mathbf{u}, \mathbf{u} \rangle = 0 \implies \langle \mathbf{y}, \mathbf{u} \rangle - \alpha\langle \mathbf{u}, \mathbf{u} \rangle = 0$$

$$\alpha = \frac{\langle \mathbf{y}, \mathbf{u} \rangle}{\langle \mathbf{u}, \mathbf{u} \rangle} = \frac{\mathbf{y}^T\mathbf{u}}{||\mathbf{u}||^2}$$

Theorem 4.1. *(Best approximation theorem) Let W a subspace with dimension p of $\mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^n$ and $\hat{\mathbf{y}} = \text{proj}_W(\mathbf{y})$. Then $||\mathbf{y} - \hat{\mathbf{y}}|| < ||\mathbf{y} - \mathbf{v}||$, $\forall \mathbf{v} \in W$, $\mathbf{v} \neq \hat{\mathbf{y}}$.*

This means that $\hat{\mathbf{y}}$ is the best approximation of $\mathbf{y}$ in W.

5 Singular Value Decomposition

Any matrix $\mathrm{A} \in \mathbb{R}^{m \times n},\ m > n$ with $\mathrm{rank}(\mathrm{A}) = k,\ k < n$ can be decomposed as

$$\mathrm{A} = \mathrm{U}\Sigma\mathrm{V}^T$$

where

- $\mathrm{U} \in \mathbb{R}^{m \times m}$ is an orthogonal matrix whose columns are the eigenvectors of $\mathrm{A}\mathrm{A}^T$ and are called *left-singular vectors*
- $\mathrm{V} \in \mathbb{R}^{n \times n}$ is an orthogonal matrix whose columns are the eigenvectors of $\mathrm{A}^T\mathrm{A}$ and are called *right-singular vectors*
- $\Sigma \in \mathbb{R}^{m \times n}$ is a diagonal matrix whose diagonal entries are the *singular values* σ_i of A, with $\sigma_i = \sqrt{\lambda_i(\mathrm{A}^T\mathrm{A})},\ i = \ldots, n$. The singular values enjoy the following property

  $$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k > \sigma_{k+1} = \sigma_{k+2} = \ldots = \sigma_n = 0$$

  where $k = \mathrm{rank}(\mathrm{A})$.

This decomposition is called *singular value decomposition* (SVD).

5.1 Moore-Penrose inverse

The SVD can be used to compute the *Moore-Penrose inverse* (or simply *pseudoinverse*) of a matrix $\mathrm{A} \in \mathbb{R}^{m \times n}$

$$\mathrm{A}^+ = \mathrm{V}\Sigma^+\mathrm{U}^T$$

where $\Sigma^+ \in \mathbb{R}^{n \times m}$ is the pseudoinverse of Σ, computed by taking the reciprocal of every non-zero diagonal element, leaving the zeros in place and transposing the matrix

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \ldots & 0 \\ 0 & \sigma_2 & \ldots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sigma_n \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \Sigma^+ = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & \ldots & 0 & \ldots & 0 \\ 0 & \frac{1}{\sigma_2} & \ldots & 0 & \ldots & 0 \\ 0 & 0 & \ddots & 0 & \ldots & 0 \\ 0 & 0 & 0 & \frac{1}{\sigma_n} & \ldots & 0 \end{bmatrix}$$

5.2 Notes on the condition number

We have that

$$\sigma_1 = \sqrt{\lambda_{max}(\mathrm{A}^T\mathrm{A})} = \sqrt{\rho(\mathrm{A}^T\mathrm{A})} = ||A||_2$$

$$\sigma_n = \sqrt{\lambda_{min}(\mathrm{A}^T\mathrm{A})} \implies ||\mathrm{A}^{-1}|| = \frac{1}{\sigma_n}$$

Thus,

$$K(\mathrm{A}) = \frac{\sigma_1}{\sigma_2}$$

5.3 Matrix approximation using SVD

Given the SVD of a matrix $\mathrm{A} \in \mathbb{R}^{m\times n}$

$$\mathrm{A} = \mathrm{U}\Sigma\mathrm{V}^T$$

we can use it to represent the matrix A as a sum of matrices $\mathrm{A}_i \in \mathbb{R}^{m\times n}$ with $\mathrm{rank}(\mathrm{A}_i) = 1$ such that

$$\mathrm{A}_i = \mathbf{u}_i{\mathbf{v}_\mathrm{i}}^T$$

The matrix A of $\mathrm{rank}(\mathrm{A}) = k$ can then be written as

$$\mathrm{A} = \sum_{i=1}^{k} \sigma_i \mathrm{A}_i = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i{\mathbf{v}_\mathrm{i}}^T$$

To obtain a rank-p approximation ($p < k$) of A we can truncate the sum at index $i = p$. The error introduced with this approximation can be computed as

$$||\mathrm{A} - \mathrm{A}_p||_2 = \left|\left| \sum_{i=p+1}^{k} \sigma_i \mathrm{A}_i \right|\right|_2 = \sigma_{p+1}$$

This means that if σ_{p+1} is small we have a good approximation of the original matrix A.

Theorem 5.1. *Given* $\mathrm{A} \in \mathbb{R}^{m\times n}$, $\mathrm{rank}(\mathrm{A}) = k$, *let* $\mathrm{A} = \mathrm{U}\Sigma\mathrm{V}^T$ *be its singular value decomposition. Let* $\mathrm{A}_p = \sum_{i=1}^{p} \sigma_i \mathbf{u}_i{\mathbf{v}_\mathrm{i}}^T$ *be the rank-p approximation of* A. *Then*

$$\forall \mathrm{B} \in \mathbb{R}^{m\times n},\ \mathrm{rank}(\mathrm{B}) = p,\ ||\mathrm{A} - \mathrm{A}_p||_2 \leq ||\mathrm{A} - \mathrm{B}||_2$$

So A_p *is the best rank-p approximation of* A *obtained via singular value decomposition.*

6 Linear Least Squares Problem

Consider an *overdetermined system*

$$\mathrm{A}\mathbf{x} = \mathbf{b}$$

where A has dimension $m \times n$ with $m > n$, $\mathbf{x}$ of dimension n and $\mathbf{b}$ of dimension m. Such a system usually has no solution, meaning that $\mathbf{b}$ does not lie on the subspace spanned by the columns of A (denoted by Col(A) from now on). Since no exact solution exists for this problem, we would like to compute the best approximate solution.

The idea is to use projections to find the vector in Col(A) that is closest to $\mathbf{b}$. As already seen in the previous chapter, this vector is indeed the orthogonal projection of $\mathbf{b}$ onto the subspace Col(A).

6.1 Existence and uniqueness of the solution

Two different cases are possible:

– $\mathrm{rank}(\mathrm{A}) = n$, meaning that every column of A is linearly independent. A unique solution exists $\forall \mathbf{b} \in \mathbb{R}^m$. Let $\mathbf{x}^*$ be the approximate solution to the linear system, we have

$$\mathrm{A}\mathbf{x}^* = \mathrm{proj}_{\mathrm{Col(A)}}(\mathbf{b})$$

This means that

$$\mathbf{b} - \mathrm{A}\mathbf{x}^* \in [\mathrm{Col(A)}]^{\perp} \implies \mathrm{A}^T(\mathbf{b} - \mathrm{A}\mathbf{x}^*) = 0$$

Thus, we obtain the *normal equation*

$$\mathrm{A}^T\mathrm{A}\mathbf{x}^* = \mathrm{A}^T\mathbf{b}$$

which is a linear system in n equations and n unknowns. $\mathrm{A}^T\mathrm{A}$ is called *Gramian matrix* of A and is non-singular, positive definite and symmetric. The approximate solution can then be obtained

$$\mathbf{x}^* = (\mathrm{A}^{\mathrm{T}}\mathrm{A})^{-1}\mathrm{A}^T\mathbf{b}$$

– rank(A) $= k$, $k < n$, meaning that there is an infinite number of solutions (∞^{n-k}). We can find the least squares solution to the problem by using the pseudoinverse of A obtained via SVD

$$\mathrm{A}\mathbf{x} = \mathbf{b} \implies \mathrm{A}^{+}\mathrm{A}\mathbf{x}^{*} = \mathrm{A}^{+}\mathbf{b} \implies \mathbf{x}^{*} = \mathrm{A}^{+}\mathbf{b} \implies \mathbf{x}^{*} = \sum_{i=1}^{k} \frac{\mathbf{u}_{\mathrm{i}}{}^{T}\mathbf{b}}{\sigma_i}\mathbf{v}_i$$

7 Multivariate Calculus

7.1 Partial derivatives

Definition 7.1. Let $f : \mathbb{R}^n \to \mathbb{R}$. We say that f is *differentiable* with respect to x_i if the limit

$$\lim_{h\to 0} \frac{f(x_1, \ldots, x_i + h, \ldots, x_n) - f(x_1, \ldots, x_n)}{h} = \frac{\partial f}{\partial x_i}(\mathbf{x})$$

exists.

Definition 7.2. A function $f : \mathbb{R}^n \to \mathbb{R}$ is said differentiable at a point $\mathbf{x}_0 \in \mathbb{R}^n$ iff all the partial derivatives of f exists at $\mathbf{x}_0$.

7.1.1 Differentiation rules

– *product rule*:

$$(fg)' = f'g + fg'$$

– *quotient rule*:

$$\left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2}$$

– *chain rule*:

$$[g(f)]' = g'(f) \cdot f'$$

7.2 Gradient

Definition 7.3. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function. Then

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}(\mathbf{x}), \ldots, \frac{\partial f}{\partial x_n}(\mathbf{x})\right)$$

is called *gradient* of f.

7.3 Hessian

Definition 7.4. Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable up to the second order. We call *Hessian* the following matrix

$$\mathrm{H}_f(\mathbf{x}) = \nabla^2 f(\mathbf{x}) = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1 \partial x_n} \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f}{\partial x_n \partial x_1} & \dfrac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

7.4 Jacobian

Definition 7.5. Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ be a function such that all its first-order partial derivatives exists. We call *Jacobian* the following matrix

$$\mathrm{J}_f(\mathbf{x}) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}) = \begin{bmatrix} \nabla f_1(\mathbf{x}) \\ \nabla f_2(\mathbf{x}) \\ \vdots \\ \nabla f_m(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} & \cdots & \dfrac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial f_m}{\partial x_1} & \dfrac{\partial f_m}{\partial x_2} & \cdots & \dfrac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Proposition 7.1. Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ and $\mathbf{g} : \mathbb{R}^m \to \mathbb{R}^p$. The Jacobian matrix of the function $\mathbf{g} \circ \mathbf{f} : \mathbb{R}^n \to \mathbb{R}^p$ is given by

$$\mathrm{J}_{\mathbf{g}\circ\mathbf{f}}(\mathbf{x}) = \mathrm{J}_{\mathbf{g}}(\mathbf{f}(\mathbf{x})) \cdot \mathrm{J}_{\mathbf{f}}(\mathbf{x})$$

Example 7.1. Let $F : \mathbb{R}^3 \to \mathbb{R}^2$ and $G : \mathbb{R}^2 \to \mathbb{R}^2$ defined as follows:

$$F(u, v, w) = (u + v, e^w) \quad \forall (u, v, w) \in \mathbb{R}^3$$

$$G(x,y) = (y - x, \sin(2\pi x)) \quad \forall (x,y) \in \mathbb{R}^2$$

The Jacobian matrix of the composition $G \circ F$ is

$$\mathrm{J}_{G\circ F}(u,v,w) = \mathrm{J}_G(F(u,v,w)) \cdot \mathrm{J}_F(u,v,w) =$$

$$= \begin{bmatrix} -1 & 1 \\ 2\pi\cos(2\pi x) & 0 \end{bmatrix}\Bigg|_{x=u+v, y=e^w} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & e^w \end{bmatrix} =$$

$$= \begin{bmatrix} -1 & -1 & e^w \\ 2\pi\cos(2\pi(u+v)) & 2\pi\cos(2\pi(u+v)) & 0 \end{bmatrix}$$

7.5 Chain rule

The *chain rule* is a formula to compute the derivative of a composite function. That is, if f and g are differentiable functions, then the chain rule expresses the derivative of their composite $F = f \circ g$ in terms of the derivatives of f and g as follows (in Lagrange's notation):

$$F'(x) = f'(g(x))g'(x)$$

If a variable z depends on a variable y, which itself depends on a variable x, the chain rule may be rewritten in Leibniz's notation as:

$$\frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx}$$

or more precisely:

$$\left.\frac{dz}{dx}\right|_x = \left.\frac{dz}{dy}\right|_{y(x)} \cdot \left.\frac{dy}{dx}\right|_x$$

Example 7.2. Let $f(x) = \sin(x)$ and $x(t) = t^2$. We have $f(t) = \sin t^2$. To compute the derivative of f with respect to t we apply the chain rule:

$$\frac{df}{dt} = \frac{df}{dx} \cdot \frac{dx}{dt} = \cos(x)|_{x=t^2} \cdot 2t = 2t\cos(t^2)$$

The chain rule is a key tool in *automatic differentiation* (AD) (also called *algorithmic differentiation*, or *computational differentiation*), which is a set of techniques to numerically evaluate the derivative of a function specified by a computer program. It exploits the fact that any arbitrary order derivative

can be rewritten via the chain rule in a sequence of elementary arithmetic operations that can be executed by every computer program.

For the simple composition

$$\begin{gathered} y = f(g(h(x))) = f(g(h(w_0))) = f(g(w_1)) = f(w_2) = w_3 \\ w_0 = x \\ w_1 = h(w_0) \\ w_2 = g(w_1) \\ w_3 = f(w_2) = y \end{gathered}$$

the chain rule gives

$$\frac{dy}{dx} = \frac{dy}{dw_2} \cdot \frac{dw_2}{dw_1} \cdot \frac{dw_1}{dx}$$

Usually, two distinct modes of AD are presented, *forward accumulation* (or *forward mode*) and *reverse accumulation* (or *reverse mode*). Forward accumulation specifies that one traverses the chain rule from inside to outside (that is, first compute $\frac{dw_1}{dx}$ and then $\frac{dw_2}{dw_1}$ and at last $\frac{dy}{dw_2}$), while reverse accumulation has the traversal from outside to inside (first compute $\frac{dy}{dw_2}$ and then $\frac{dw_2}{dw_1}$ and at last $\frac{dw_1}{dx}$). More succinctly,

- *forward mode* computes the recursive relation $\frac{dw_i}{dx} = \frac{dw_i}{dw_{i-1}} \cdot \frac{dw_{i-1}}{dx}$ with $w_3 = y$
- *forward mode* computes the recursive relation $\frac{dy}{dw_i} = \frac{dy}{dw_{i+1}} \cdot \frac{dw_{i+1}}{dw_i}$ with $w_0 = x$

Example 7.3. Consider the function

$$\begin{aligned} z = f(x_1, x_2) = \\ = x_1 x_2 + \sin(x_1) = \\ = w_1 w_2 + \sin(w_1) = \\ = w_3 + w_4 = \\ = w_5 \end{aligned}$$

Forward mode automatic differentiation We want to compute the gradient of z using forward mode automatic differentiation. We will have to perform two sweeps over the computational graph since we need to differentiate with respect to both x_1 and x_2. The choice of the independent variable to which differentiation is performed affects the seed values, which are

$$\frac{dw_1}{dx_1} = 1, \quad \frac{dw_2}{dx_1} = 0$$

if we differentiate with respect to x_1, and

$$\frac{dw_1}{dx_2} = 0, \quad \frac{dw_2}{dx_2} = 1$$

if we differentiate with respect to x_2 With the seed values set, the values propagate as follows (choosing x_1 as independent variable):

$$\begin{aligned}
\frac{dw_1}{dx_1} &= 1 \\
\frac{dw_2}{dx_1} &= 0 \\
\frac{dw_3}{dx_1} &= \frac{dw_1}{dx_1} w_2 + w_1 \frac{dw_2}{dx_1} = w_2 \\
\frac{dw_4}{dx_1} &= \frac{dw_4}{dw_1} \cdot \frac{dw_1}{dx_1} = \cos(w_1) \cdot 1 = \cos(w_1) \\
\frac{dw_5}{dx_1} &= \frac{dw_3}{dx_1} + \frac{dw_4}{dx_1} = w_2 + \cos(w_1)
\end{aligned}$$

Reverse mode automatic differentiation In reverse mode automatic differentiation, the dependent variable to be differentiated is fixed and the derivative is computed with respect to each sub-expression recursively. This means that our seed is

$$\frac{dz}{dw_5} = 1$$

and the operations are computed as follows:

$$\frac{dz}{dw_5} = 1$$
$$\frac{dz}{dw_4} = \frac{dz}{dw_5} \cdot \frac{dw_5}{dw_4} = 1 \cdot 1 = 1$$
$$\frac{dz}{dw_3} = \frac{dz}{dw_5} \cdot \frac{dw_5}{dw_3} = 1 \cdot 1 = 1$$
$$\frac{dz}{dw_2} = \frac{dz}{dw_3} \cdot \frac{dw_3}{dw_2} = 1 \cdot w_1 = w_1$$
$$\frac{dz}{dw_1} = \frac{dz}{dw_3} \cdot \frac{dw_3}{dw_1} + \frac{dz}{dw_4} \cdot \frac{dw_4}{dw_1} = 1 \cdot w_2 + 1 \cdot \cos(w_1) = w_2 + \cos(w_1)$$

8 Probability and Statistics

Definition 8.1. A *random experiment* is an experiment whose outcome is determined by chance (e.g. roll of a die).

Definition 8.2. The *sample space* is the set of all possible results of a random experiment.

Definition 8.3. An *event* is a collection of results. It is also defined as a subset of the sample space.

Definition 8.4. The *space of the events* $S(A)$ is the set of all the possible events.

Definition 8.5. The *probability of an event* A is a function $P : S(A) \to [0, 1] \in \mathbb{R}$ that associates each event A to a number called probability of A.

Proposition 8.1. Each probability function P satisfies:

- $P(A) \geq 0$
- $P(S) = 1$
- if $A_1, A_2, \ldots, A_n$ are disjoint events ($A_1 \cap A_2 \cap \ldots \cap A_n = \emptyset$) then

$$P(A_1 \cup A_2 \cup \ldots \cup A_n) = \sum_{i=1}^{n} P(A_i)$$

Definition 8.6. The *conditional probability* of an event B given the event A is defined as

$$P(B \mid A) = \frac{P(A \cup B)}{P(A)}$$

Proposition 8.2. For any fixed A, with $P(A) > 0$:

- $P(B \mid A) \geq 0, \forall B \subset S$
- $P(S \mid A) = 1$
- $B_1, B2, \ldots, B_n$ disjoint events

$$P\left(\bigcup_{i=1}^{n} B_i \mid A\right) = \sum_{i=1}^{n} P(B_i \mid A)$$

- $\forall A_1, A_2, \ldots, A_n$ events:

$$P(A_1 \cap \ldots \cap A_n) = P(A) \cdot P(A_2 \mid A_1) \cdot \ldots \cdot P(A_n \mid A_1 \cap \ldots \cap A_{n-1})$$

Definition 8.7. Two events A and B are *independent* iff

$$P(A \cap B) = P(A) \cdot P(B)$$

Two events which are not independent are called *dependent*.

Theorem 8.1. *Given A, B disjoint events, $A \cup B = S$, we have that*

$$P(B \mid A) = \frac{P(A \mid B)P(B)}{P(A)}$$

The theorem can be extended to multiple events $B_1, \ldots, B_n$ pairwise disjoint ($B_i \cap B_j = \emptyset,\ \forall i \neq j$) and exhaustive ($B_1 \cup \ldots \cup B_n = S$)

$$P(B_i \mid A) = \frac{P(A \mid B_i)P(B_i)}{\sum_{i=j}^{n} P(A \mid B_j)P(B_j)}$$

8.1 Random variables

Definition 8.8. A *random variable* is a function $X : S \to \mathbb{R}$ that associates each outcome $w \in S$ to a number $x \in \mathbb{R}$

$$X(w) = x$$

Definition 8.9. The set of all the possible values of a random variable X is called *support of* X, S_X.

If the target space (support) is a countable set then the random variable is called *discrete*.

If the target space (support) is a non countable set then the random variable is called *continuous*.

8.1.1 Discrete random variables

Each discrete random variable has a function associated to itself, called *probability mass function* (PMF)

$$f_X : S_X \to [0, 1] \in \mathbb{R}$$

such that

$$f_X(x) = P(X = x),\ x \in S_X$$

and

$$\sum_{x \in S_X} f_X(x) = 1$$

Definition 8.10. The *expectation* of a discrete random variable is defined as

$$\mu = \mathbb{E}[X] = \sum_{x \in S_X} x f_X(x)$$

The expectation of a function $g(X)$ is defined as

$$\mathbb{E}[g(X)] = \sum_{x \in S_X} g(x) f_X(x)$$

Definition 8.11. The *variance* of a discrete random variable is defined as

$$\sigma^2 = \mathbb{E}[(X - \mu)^2] = \sum_{x \in S_X} (x - \mu)^2 f_X(x) = \mathbb{E}[X^2] - \mathbb{E}^2[X]$$

Some examples of probability mass function are:

– the *discrete uniform distribution* (n is the dimension of the support)

$$f_X(x) = \frac{1}{n}$$

– the *Poisson distribution*

$$f_X(x) = e^{-\lambda}\frac{\lambda^x}{x!}$$

which has $\mu = \lambda$ and $\sigma^2 = \lambda$

8.2 Continuous random variables

Each discrete random variable has a function associated to itself, called *probability density function* (PDF)

$$f_X : S_X \to [0, 1] \in \mathbb{R}$$

such that

$$P(a \leq X \leq b) = \int_a^b f_X(x)dx, \ \forall [a, b] \in S_X$$

and

$$\int_{x \in S_X} f_X(x) = 1$$

Definition 8.12. The *expectation* of a continuous random variable is defined as

$$\mu = \mathbb{E}[X] = \int_{x \in S_X} x f_X(x)$$

The expectation of a function $g(X)$ is defined as

$$\mathbb{E}[g(X)] = \int_{x \in S_X} g(x) f_X(x)$$

Definition 8.13. The *variance* of a continuous random variable is defined as

$$\sigma^2 = \mathbb{E}[(X - \mu)^2] = \int_{x \in S_X} (x - \mu)^2 f_X(x) = \mathbb{E}[X^2] - \mathbb{E}^2[X]$$

Some examples of probability mass function are:

– the *normal distribution* (or *Gaussian distribution*)

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

with expectation μ and standard deviation σ. The *standard normal distribution* is defined with parameters $\mu = 0$ and $\sigma = 1$.

– the *exponential distribution*

$$f_X(x) = \lambda e^{-\lambda x}$$

which has mean $\mu = \frac{1}{\lambda}$ and standard deviation $\sigma = \frac{1}{\lambda}$

8.3 Multivariate distributions

Definition 8.14. The *joint probability mass function* of X and Y is defined as

$$f_{XY} = P(X = x, Y = y),\ (x, y) \in S_{XY}$$

where S_{XY} is the joint support set (usually $S_{XY} = S_X \times S_Y$)

Definition 8.15. Given f_{XY} joint probability mass function of X and Y we define the *marginal probability distribution* of X as

$$f_X(x) = \sum_{y \in S_Y} P(X = x, Y = y)$$

and the *marginal probability distribution* of Y as

$$f_Y(y) = \sum_{x \in S_X} P(X = x, Y = y)$$

Definition 8.16. We define the *covariance* between two random variables X and Y as

$$\mathrm{Cov}_{XY}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$$

We have $\mathrm{Cov}_{XX}(X, X) = \sigma_X^2$.

Definition 8.17. We define the *correlation* between two random variables X and Y as

$$\mathrm{Corr}_{XY}(X,Y) = \frac{\mathrm{Cov}_{XY}(X,Y)}{\sigma_X \sigma_Y}$$

If $X \in \mathbb{R}^m,\ X = (X_1, X_2, \ldots, X_m)$ and $Y \in \mathbb{R}^n,\ Y = (Y_1, Y_2, \ldots, Y_n)$ we have that $\mathrm{Cov}(X,Y)$ and $\mathrm{Corr}(X,Y)$ are matrices of size $m \times n$ such that

$$(\mathrm{Cov}(X,Y))_{ij} = \mathrm{Cov}(X_i, Y_j)$$

$$(\mathrm{Corr}(X,Y))_{ij} = \mathrm{Corr}(X_i, Y_j)$$

If $X \in \mathbb{R}^n,\ X = (X_1, X_2, \ldots, X_n)$ we have that $\mathrm{Cov}(X,X)$ is a symmetric positive definite matrix of size $n \times n$ whose diagonal terms are $\sigma^2_{X_i}$

Proposition 8.3. If X and Y are two independent random variables then

$$\mathrm{Cov}(X,Y) = 0,\ \mathrm{Corr}(X,Y) = 0$$

8.4 Inferential statistics

Inferential statistics tries to deduce underlying properties of the distribution of a population from which a sample is observed. The set of assumptions is called *statistical model* and the assumed probability distribution is usually parametrized. These parameters are then estimated to fit the observed data as well as possible.

We will study two methods to estimate the parameters of the assumed probability distribution:

- *maximum likelihood estimation* (MLE)
- *maximum a posteriori estimation* (MAP)

8.4.1 Maximum likelihood estimation

The goal of maximum likelihood estimation is to maximize the likelihood of the observed data with respect to the parameters

$$L(\theta \mid x) = f_X(x \mid \theta)$$

Here $f_X(x \mid \theta)$ is viewed as a function of the parameter θ with x fixed. This should not be seen as the probability that the parameters θ are the right ones, given the observed data.

Thus, the method of maximum likelihood estimation estimates θ as

$$\hat{\lambda}_{MLE}(x) = \arg\max_{\lambda}\{L(\theta \mid x)\} = \arg\min_{\lambda}\{-\ln L(\theta \mid x)\}$$

Example 8.1. Suppose $X_1, \ldots, X_n \sim \exp(\lambda)$ i.i.d. random variables with observations $(x_1, \ldots, x_n)$. We have

$$L(\lambda \mid x_1, \ldots, x_n) = \prod_{i=1}^{n} \lambda e^{-\lambda x_i} = \lambda^n e^{-\lambda \sum_{i=1}^{n} x_i}$$

To find λ we need to compute

$$\arg\max_{\lambda}\{\lambda^n e^{-\lambda \sum_{i=1}^{n} x_i}\} = \arg\min_{\lambda}\{-n \ln \lambda + \lambda \sum_{i=1}^{n} x_i\}$$

Let's compute the derivative of $L_1(\lambda) = -n \ln \lambda + \lambda \sum_{i=1}^{n} x_i$ with respect to λ

$$\frac{dL_1}{d\lambda}(\lambda) = -\frac{n}{\lambda} + \sum_{i=1}^{n} x_i$$

Thus, the value of λ which minimizes L_1 is $\lambda = \dfrac{n}{\sum_{i=1}^{n} x_i}$.

8.4.2 Maximum a posteriori estimation

Assume that we want to estimate an unobserved parameter θ on the basis of observations x. Let f be the sampling distribution of x, so that $f_X(x \mid \theta)$ is the probability of x when the underlying population parameter is θ. Then the function

$$\mathcal{L}(\theta \mid x) = f_X(x \mid \theta)$$

is known as likelihood function. Now suppose that a prior distribution $f_\theta(\theta)$ exists. This allows us to treat θ as a random variable. We can calculate the posterior distribution of θ using Bayes' theorem:

$$f(\theta \mid x) = \frac{f_X(x \mid \theta) f_\theta(\theta)}{f_X(x)}$$

Maximum a posteriori estimation estimates θ as

$$\hat{\theta}_{MAP}(x) = \arg\max_{\theta}\{f_X(x \mid \theta) f_\theta(\theta)\}$$

Note that the evidence $f_X(x)$ is positive and does not depend on θ, so it does not play any role in the optimization.

9 Numerical optimization

The minimization of a function of several variables can be formulated as: given $f : \mathbb{R}^n \to \mathbb{R}$, called an *objective function*,

$$\text{minimize } f(\mathbf{x}) \text{ in } \mathbb{R}^n$$

This is called an *unconstrained optimization problem.*

Tipically, we want to determine the optimal values of several variables $x_1, \ldots, x_n$ ruled by specific laws such as equality or inequality constraints. Moreover, we may require that these values lie within a subset $\Omega \subset \mathbb{R}^n$. This kind of optimization problem is called *constrained* and can be formulated as: given the objective function f,

$$\text{minimize } f(\mathbf{x}) \text{ in } \Omega \subset \mathbb{R}^n$$

In this chapter, we will address the following topics:

- conditions for the existence (and uniqueness) of a solution
- numerical algorithms to solve these kinds of problems

Definition 9.1. Let $f : \mathbb{R}^n \to \mathbb{R}$. $\mathbf{x}^* \in \mathbb{R}^n$ is called a *local minimum point* (resp. *strict local minimum point*) of f if there exists $\epsilon > 0$ such that

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \text{ (resp. } f(\mathbf{x}^*) < f(\mathbf{x})), \quad \forall ||\mathbf{x} - \mathbf{x}^*|| < \epsilon$$

Definition 9.2. Let $f : \mathbb{R}^n \to \mathbb{R}$. $\mathbf{x}^* \in \mathbb{R}^n$ is called a *global minimum point* (resp. *strict global minimum point*) of f if

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \text{ (resp. } f(\mathbf{x}^*) < f(\mathbf{x})), \quad \forall \mathbf{x} \in \mathbb{R}^n, \ \mathbf{x} \neq \mathbf{x}^*$$

Theorem 9.1. *(First order optimality condition, also called Fermat's theorem for stationary points) Let $f : \mathbb{A} \in \mathbb{R}$, with $\mathbb{U} \subseteq \mathbb{R}^n$ open set. If $\mathbf{x}^* \in \mathbb{U}$ is a local optimum point for f and f is differentiable in $\mathbf{x}^*$, then*

$$\nabla f(\mathbf{x}^*) = 0$$

Note that this is a necessary condition.

Theorem 9.2. *(Second order optimality condition) Let $f : \mathbb{A} \in \mathbb{R}$, with $\mathbb{U} \subseteq \mathbb{R}^n$ open set. If $\mathbf{x}^* \in \mathbb{U}$ is a local minimum point for f and f is twice differentiable around $\mathbf{x}^*$, then*

$$\nabla f(\mathbf{x}^*) = 0 \text{ and } \nabla^2 f(\mathbf{x}^*) \text{ is positive semidefinite}$$

Note that this is a necessary condition.

Theorem 9.3. *If $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable with continuity around $\mathbf{x}^* \in \mathbb{R}^n$ and $\nabla f(\mathbf{x}^*) = 0$ and $\nabla^2 f(\mathbf{x}^*)$ is positive definite, then $\mathbf{x}^*$ is a strict local minimum point.*

Definition 9.3. $\mathbb{C} \subset \mathbb{R}^n$ is a *convex set* if $\forall \mathbf{x}, \mathbf{y} \in \mathbb{C}$ and $\theta \in [0, 1]$ we have

$$\theta\mathbf{x} + (1 - \theta)\mathbf{y} \in \mathbb{C}$$

Definition 9.4. Let $f : \mathbb{D} \to \mathbb{R}$, with $D \subset \mathbb{R}^n$ convex set. Then f is a *convex function* if $\forall \mathbf{x}, \mathbf{y} \in \mathbb{D}$, $\theta \in [0, 1]$ we have

$$f(\theta x + (1 - \theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1 - \theta) f(\mathbf{y})$$

Proposition 9.1. We have the following:

- If f is twice differentiable in $\mathbf{x}$ and convex (resp. strictly convex), then $\nabla^2 f(\mathbf{x})$ is positive semidefinite (resp. positive definite)
- If f is a convex function then each point of local minimum is a point of global minimum
- If f is strictly convex then there exists a unique point of global minimum

9.1 Convex quadratic functions

Convex quadratic functions take the following form:

$$q(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \mathrm{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x}\ (+\mathbf{w})$$

with $\mathbf{x}, \mathbf{c} \in \mathbb{R}^n$ and $\mathrm{Q} \in \mathbb{R}^{n\times n}$ symmetric positive definite. A typical quadratic optimization problem is the least square problem

$$\min ||\mathrm{A}\mathbf{x} - \mathbf{b}||^2$$

The objective is to minimize

$$\begin{aligned}
\frac{1}{2}||\mathrm{A}\mathbf{x} - \mathbf{b}||^2 &= \frac{1}{2}(\mathrm{A}\mathbf{x} - \mathbf{b})^T(\mathrm{A}\mathbf{x} - \mathbf{b}) = \\
&= \frac{1}{2}(\mathbf{x}^T\mathrm{A}^T - \mathbf{b}^T)(\mathrm{A}\mathbf{x} - \mathbf{b}) = \\
&= \frac{1}{2}(\mathbf{x}^T\mathrm{A}^T\mathrm{A}\mathbf{x} - \mathbf{x}^T\mathrm{A}^T\mathbf{b} - \mathbf{b}^T\mathrm{A}\mathbf{x} + \mathbf{b}^T\mathbf{b}) = \\
&= \frac{1}{2}\mathbf{x}^T\mathrm{A}^T\mathrm{A}\mathbf{x} - \mathbf{b}^T\mathrm{A}\mathbf{x} + \frac{1}{2}\mathbf{b}^T\mathbf{b}
\end{aligned}$$

which can be rewritten in quadratic form by setting $\mathrm{Q} = \mathrm{A}^T\mathrm{A}$ and $\mathbf{c}^T = -\mathbf{b}^T\mathrm{A}$ We have

$$\nabla f(\mathbf{x}) = \mathrm{A}^T\mathrm{A}\mathbf{x} - \mathrm{A}^T\mathbf{b}$$

and setting $\nabla f(\mathbf{x}) = 0$ we obtain the normal equation

$$(\mathrm{A}^T\mathrm{A})\mathbf{x} = \mathrm{A}^T\mathbf{b}$$

9.2 Iterative methods

Iterative methods can be formulated as follows: given an initial vector $\mathbf{x}_0 \in \mathbb{R}^n$, compute for $k \geq 0$ until convergence

$$\mathbf{x}_{k+1} = g(\mathbf{x}_k)$$

where g is an arbitrary function. We have that $\mathbf{x}_k \to \mathbf{x}^*$ for $k \to \infty$, where $\mathbf{x}^*$ is a stationary point (local minimum).

Since, for obvious reasons, we can't compute an infinite number of terms, we need to set a *stopping criterion*. Some examples are:

– *absolute criterion* based on first order condition, $||\nabla f(\mathbf{x}_k)|| < \tau_A$, where τ_A is the chosen tolerance

– *relative criterion*, $\dfrac{||\nabla f(\mathbf{x}_k)||}{||f||(\mathbf{x}_0)} < \tau_R$, where τ_R is the chosen tolerance

– *absolute criterion* on succeeding values, $||\mathbf{x}_{k+1} - \mathbf{x}_k|| < \tau_{AP}$, where τ_{AP} is the tolerance

– *relative criterion* on succeeding values, $\dfrac{||\mathbf{x}_{k+1} - \mathbf{x}_k||}{\mathbf{x}_k} < \tau_{RP}$, where τ_{RP} is the tolerance

9.3 Descent methods

Descent methods are iterative methods that can be formulated as follows: given an initial vector $\mathbf{x}_0 \in \mathbb{R}^n$, compute for $k \geq 0$ until convergence

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

where $\mathbf{p}_k$ is a suitably chosen descent direction and α_k is a positive parameter called *stepsize* that measures the step along the direction $\mathbf{p}_k$. This direction is called a *descent direction* if

$$\begin{aligned} &\mathbf{p}_k^T \nabla f(\mathbf{x}_k) < 0 \quad &&\text{if } \nabla f(\mathbf{x}_k) \neq 0 \\ &\mathbf{p}_k = 0 &&\text{if } \nabla f(\mathbf{x}_k) = 0 \end{aligned}$$

provided that f is continuously differentiable around $\mathbf{x}_k$. This ensures that there exists $\alpha_k > 0$, sufficiently small, such that

$$f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) < f(\mathbf{x}_k)$$

The choice of $\mathbf{p}_k$ corresponds to different methods.

9.3.1 Gradient method (steepest descent method)

The direction is set as $\mathbf{p}_k = -\nabla f(\mathbf{x}_k)$, thus

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)$$

9.3.2 Newton's method

The direction is set as

$$\mathbf{p}_k = -\mathrm{H}_f^{-1}(\mathbf{x}_k)\nabla f(\mathbf{x}_k)$$

provided that H_f is positive definite within a sufficiently large neighborhood of $\mathbf{x}^*$. At each iteration $\mathbf{p}_k$ is computed as a solution of the following linear system

$$\mathrm{H}_f(\mathbf{x}_k)\mathbf{p}_k = \nabla f(\mathbf{x}_k)$$

9.3.3 Inexact Newton's methods

Theorem 9.4. *If f is twice differentiable with continuity in its domain and is a Lipschitz function, that is*

$$\exists L > 0 \text{ such that } ||\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})|| \leq L||\mathbf{x} - \mathbf{y}||$$

, and H_f is positive definite in a neighbour of the minimum, then

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathrm{H}_f^{-1}(\mathbf{x}_k)\nabla f(\mathbf{x}_k)$$

converges to a stationary point.

This lets us approximate $\mathrm{H}_f(\mathbf{x}_k)$ with a positive definite matrix $\mathrm{B}_f(\mathbf{x}_k)$.

$$\mathrm{H}_f(\mathbf{x}_k) \simeq \mathrm{B}(\mathbf{x}_k)$$

9.4 Line search techniques

Line search techniques are methods used to find admissible values of α_k. Some examples are:

- *exact line search*, tries to minimize $f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)$ with respect to α_k
- *inexact line search* (or *backtracking*), starts with an initial value for α_k and reduces it until $f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \sigma\alpha\mathbf{p}_k\nabla f(\mathbf{x}_k)$, where $\sigma \in (0, 1/2)$
- *constant*, set $\alpha_k \equiv k$

9.5 Convergence speed of a method

Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_k, \ldots\}$ be the sequence generated by a method, $\mathbf{x}^*$ a stationary point and $\lim_{k\to\infty} \mathbf{x}_k = \mathbf{x}^*$. A method can converge with:

- *Q-linear convergence speed*, if it exsists $r \in (0, 1)$ such that

$$\frac{||\mathbf{x}_{k+1} - \mathbf{x}^*||}{||\mathbf{x}_k - \mathbf{x}^*||} \leq r$$

- *Q-superlinear convergence speed*, if

$$\lim_{k\to\infty} \frac{||\mathbf{x}_{k+1} - \mathbf{x}^*||}{||\mathbf{x}_k - \mathbf{x}^*||} = 0$$

- *Q-quadratic convergence speed*, if $\forall k > k*$, $M > 0$

$$\frac{||\mathbf{x}_{k+1} - \mathbf{x}^*||}{||\mathbf{x}_k - \mathbf{x}^*||^2} < M$$

Gradient methods have a Q-linear convergence speed (slower), while Newton-like methods have a Q-quadratic convergence speed (faster), which compensates their higher computational cost.

9.6 Regularization

Regularization techniques can be used to slightly change ill-conditioned matrices. The simplest method is to add a diagonal matrix multiplied by a *regularization parameter*, λI. Regularization also attempts to impose Occam's razor on the solution, meaning that it helps generate a simpler solution which is less affected by noise. For example, in Baesyan statistics we can impose certain prior distributions (regularization term) on model parameters

$$\hat{\theta}_{MAP} = -\arg\min_{\theta} \sum_{i=1}^{n} \log f_X(x_i \mid \theta) + \lambda \log p(\theta)$$

9.7 Non-linear least squares problem

In non-linear least squares problems we have residuals defined as $\mathbf{r} : \mathbb{R}^n \to \mathbb{R}^m$, where n is the dimension of the space of the data and m is the dimension of the space of the unknowns. Each $r_i(\mathbf{x})$ is a non-linear function and the minimization of the residuals can be written as

$$\min ||\mathbf{r}(\mathbf{x})||^2 = \min \sum_{i=1}^{m} r_i^2(\mathbf{x})$$

We can write the gradient of $f : \mathbb{R}^m \to \mathbb{R}$, $f = \frac{1}{2} \sum_{i=1}^{m} r_i^2(\mathbf{x})$ as

$$\nabla f(\mathbf{x}) = \sum_{i=1}^{m} r_i(\mathbf{x}) \nabla r_i(\mathbf{x}) = \mathrm{J}_r^T(\mathbf{x}) \cdot \mathbf{r}(\mathbf{x})$$

and the hessian of f as

$$\mathrm{H}_f(\mathbf{x}) = \sum_{i=1}^{m} \left(\nabla r_i(\mathbf{x}) \cdot \nabla^T r_i(\mathbf{x}) + \nabla r_i(\mathbf{x}) \cdot \mathrm{H}_{r_i}^T(\mathbf{x}) \right) =$$
$$= \mathrm{J}_r^T(\mathbf{x}) \cdot \mathrm{J}_r(\mathbf{x}) + \sum_{i=1}^{m} \nabla r_i(\mathbf{x}) \cdot \mathrm{H}_{r_i}^T(\mathbf{x})$$

These formulas can be very expensive to compute. The methods used to solve these problems are:

- *gradient methods*
- *Newton-like methods*
- *Gauss-Newton method*, where $\mathbf{p}_k$ is computed as

$$(\mathrm{J}_r^T \cdot \mathrm{J}_r)(\mathbf{x}_k) \cdot \mathbf{p}_k = -\mathrm{J}_r^T(\mathbf{x}_k) \cdot \mathbf{r}(\mathbf{x}_k)$$

- *Levenberg-Marquardt method*, adds regularization because $(\mathrm{J}_r^T \cdot \mathrm{J}_r)$ can be ill-conditioned. $\mathbf{p}_k$ is computed as

$$(\mathrm{J}_r^T(\mathbf{x}_k) \cdot \mathrm{J}_r(\mathbf{x}_k) + \lambda \mathrm{I}) \cdot \mathbf{p}_k = -\mathrm{J}_r^T(\mathbf{x}_k) \cdot \mathbf{r}(\mathbf{x}_k)$$

9.8 Stochastic optimization

The objective function in some optimization problems can take the following form

$$G(\mathbf{x}) = \sum_{i=1}^{n} G_i(\mathbf{x})$$

with $G : \mathbb{R}^n \to \mathbb{R}$. Each G_i is tipically associated with the i-th observation in the dataset.

In the case of standard (or *batch*) gradient descent, the iteration is

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - \alpha_k \nabla G(\mathbf{x}_k) \\ &= \mathbf{x}_k - \alpha_k \sum_{i=1}^{n} \nabla G_i(\mathbf{x}_k) \end{aligned}$$

which can be very expensive to evaluate.

In *stochastic gradient descent* the true gradient $\nabla G(\mathbf{x}_k)$ is approximated at each iteration by the gradient at a single observation $\nabla G_i(\mathbf{x}_k)$, with $i = 1, \ldots, n$ randomly picked. The iteration step becomes

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla G_i(\mathbf{x}_k)$$

A compromise between batch gradient descent and stochastic gradient descent is to compute the gradient against a set (called *mini-batch*) of randomly picked observations. This method is therefore called *mini-batch stochastic gradient descent*. The iteration step becomes

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \sum_{i \in M} \nabla G_i(\mathbf{x}_k), \quad M \subset \{1, \ldots, n\}$$

Stochastic gradient descent is less computationally demanding with respect to batch gradient descent, but its convergence speed is lower. The convergence of stochastic gradient descent has been analyzed using the theories behind convex minimization and stochastic approximation.